**Day 1 Agenda and Total duration is 3:00:13**

* Legacy Systems - Need for a change
* Introduction to Cloud computing
* Advantages of Cloud computing
* Capex vs Opex for Business
* Service models
* Deployment Models
* Shared Responsibility Model
* AWS DataCenters
* AWS Console Tour

**Creating an AWS Account**

* Google AWS Management Console
* SignIn to the console
* Create a free tier account and create a new account
* Select the Basic Plan and create an account

**Cloud Computing?**

**Cloud computing** is the delivery of different services through the Internet. These resources include tools and applications like data storage, servers, databases, networking, and software.

**Cloud providers**

* AWS - Amazon -First in Cloud shares
* Azure - Microsoft Second in Cloud Shares
* Google cloud
* IBM
* VMWare
* Alibaba cloud

**Every Corporate company wants to move its resources/products safely into the cloud to optimize the investment.**

**The need for Cloud computing**

**Problem with the legacy system.**

For example, I have sensitive data that is stored within a pen drive for a  presentation and somehow I lost that. When I lose my Pendrive there were so many security issues. Here the people can access my data and it is a big breach

For example, if you have your data in a google drive, you can access the data anywhere from the globe. It is giving us portable access where cloud computing is coming into the picture.

**Physical Infrastructure:**

For example, Your university has declared the results. On that day many of you are experienced with this common problem “Server is down”.

For you, this might be a website but in the background, there were servers that are running where your university hosted this application. Letus take the application of University results to be hosted in 10 servers. On a normal day, the application used by users is normal traffic.  but on the results declaration day, it may be 100 times more than the original traffic. On that day you cant add the extra servers easily because setting up a network takes a lot of background work as it requires the environmental provisioning, technical provisioning, and security measures. Let us assume you have added the servers but after the next day servers are not going to be utilized properly and wasting a lot of resources.

**What is Cloud Computing?**

A Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources(i.e, networks, servers, storage, applications, and services)

1. It can be rapidly provisioned and released with minimal management effort.
2. It provides a high-level abstraction of computation and storage

**What is the CLOUD?**

* Cloud is the ability to host a software platform or service from a remote location that can be freely accessed and used anywhere via internet access
* Cloud is a hosted service and reduces the need for physical infrastructure lowers financial investment with the “Pay as you go model”.
* The cloud supports businesses looking to optimize business processes without spending a fortune with greater accessibility, better security, and little financial outlay.

**Advantages**

* Stop guessing capacity - No need to worry about idle servers (Autoscaling)
* Increase speed and agility - The manpower and the time
* Focus on Business differentiators - The cost is getting minimized(No idle servers) by pay as you go
* Variable vs Capital Expense
* Go Global in minutes

**CAPEX vs OPEX**

|  |  |
| --- | --- |
| **CAPEX** | **OPEX** |
| Long lead times for new equipment, significant staff integration effort required | On-demand services are designed for plug and play |
| Equipment failures | Automatic instance health maintenance |
| Overprovisioning/ wasted capacity | Buy only what you need to use metrics to resize on the fly |
| Hvac, electricity facility rental insurances | none of these things are required |
| low-value maintenance staff required | cloud provider maintenance underlying hardware |
| configuration errors can be costly and difficult to remediate | software-defined services are easily configured and deliver repeatable results |
| new initiatives undergo a slow, rigid provisioning process | stand up assets as you need then, delete what you no longer need |
| significant monitoring tooling investments with Limited automation | monitoring and alerting built into each service; significant automated remediate opportunities |

**Deployment Models**

Everybody in the public cloud can have access to all the privileges that are going to be provided in the shared pool of resources but in the private cloud the people can have only some privileges

Let us consider four companies named A, B, C, and D are collaborating on a project that means they have certain files, folders, or resources that have to be shared among all the companies. Their collaborative project more specific to security. They want to have more privacy they will have more resources. private organization of four groups of people will have access to this shared pool. Company A will have the rights to set the privileges for every individual and give granular permissions

* A hybrid cloud is a combination of public and private clouds
* Public cloud has the resources which are cheaper and private cloud provides us the security

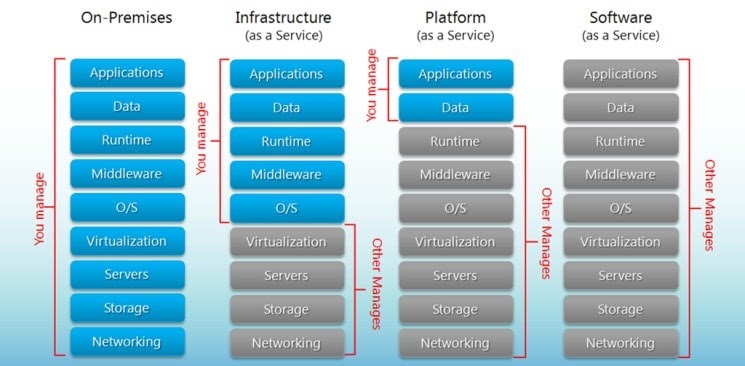
**Characteristics of public cloud**

* A public cloud is a publicly accessible cloud environment owned by a third party cloud provider
* The resources on public clouds are provided as per the service models
* The cloud providers create and managers public cloud and its resources

**Characteristics of private cloud**

* Private cloud is owned by a single organization
* Centralizing access to it is the sources from different parts, locations, or departments of the organization
* The actual administration is carried out by internal or outsourced staff
* The same organization is technically both the cloud consumer and cloud provider
* An organizational department that assumes responsibility for provisioning cloud plays the cloud provider role
* Departments requiring access to the private cloud assume the cloud consumer role
* On-premise and the cloud-based resources in a private cloud

**Service Models**



Depending on the privileges that a consumer has and the provider has service models are divided into four categories

**Infrastructure as a service**

* Provider provides and manages the virtual infrastructure and the Consumer manages the databases
* The capability provided to the consumer is to the provisioning processing storage networks and other fundamental computing resources where the consumer is able to deploy and run the software which includes OS and applications
* The consumer does not manage or control the underlying cloud infrastructure but has control over OS storage and deploy applications and Limited console of selected networking components like host firewalls

**IAAS characteristics**

* Resources are available as a service
* The cost varies depends on consumption
* Services are highly scalable
* Provides complete control of the infrastructure to organizations
* Dynamic and flexible

**When to use IAAS**

* A startup or  a small company with less capital cost for hardware and software for large organizations
* To have complete control over their applications and Infrastructures
* For Rapid growing companies you don't have to commit to specific hardware or software as your needs change and evolve ambiguity in demands for a new application

**Advantages**

* most flexible and scalable Cloud Computing model
* automated deployment of storage, networking, services, and processing power gives clients complete control of the infrastructure are the sources that can be purchased as needed

**Platform as a service**

The capability is provided to the consumer is to deploy on to the cloud infrastructure consumer-created or acquired applications created using programming languages library services and tools supported by the provider also the consumer does not manage or control over underlying cloud infrastructure including network service operating system storage but has control over the deploy applications possibly the configuration settings for the application-hosting environment

**PAAS Characteristics**

* Built on virtualization technology - easily scale up or down resources
* Assist with the development, testing, and deployment of apps
* Numerous users can access the same development application
* web services and databases are integrated

**Advantages**

* Makes the development and deployment of apps simple and cost-effective
* scalable
* highly available
* reduces the amount of coding
* Allows easy migration to a hybrid model

**When to use PAAS**

* Flexible to accommodate multiple developers working on the same development project
* To create your own customized applications
* Reduce costs for rapidly developing for deploying an app

Examples: AWS elastic beanstalk, Windows Azure, Google app engine, Apache Stratos, Open shift

**Software as a service**

* The capability provided to the consumer is to use providers applications running on a cloud infrastructure
* The applications are accessible from various client devices through a thin client interface such as a web browser or a program interface
* Also, the consumer does not manage or control over underlying cloud infrastructure including network servers os storage
* Characteristics central location hosted on a remote server accessible over the internet users not responsible for Hardware or software updates

**When to use SAAS**

* If you are a start-up or a small company that needs launcher E-Commerce quickly and don't have time for server issues or software
* For short term project that required collaboration
* If applications that are not in demand very often such as Tax software
* For applications that need both web and mobile access

Examples: Google Apps, Dropbox Salesforce GoToMeeting

**The global infrastructure of AWS**

* AWS Global infrastructure has 24 launched regions, 3 announced regions, and 77 availability zone AZ
* A region is a geographical area and its region contains two or more availability zones
* An availability zone is a data center.
* Both AZ is placed at a large distance from each other to avoid a shutdown

**Domains in AWS**

* Security Services
* Computer Services
* Storage services
* Database services
* Networking services
* Application services
* Management Services

## Day 2 Agenda and Total Duration is 3:04:45

* Elastic Compute Cloud(EC2)
* Lab Sessions
  + Creating Windows Instance
  + Creating Linus Instance

### **Elastic Compute Cloud(EC2)**

#### What is compute?

Any real-time physical instrument can be abstracted into a virtual entity(storage, networking components, or servers). All the processing can be done inside the server

1. Amazon EC2 is a web service that resizable compute capacity in the cloud.
2. It reduces the time required to obtain and boot new server instances to minutes allowing you to quickly scale capacity, both up and down as your computing requirements change
3. It changes the economics of computing by allowing you to pay only for capacity that you actually use.

#### Use Cases

* Web hosting
* Databases
* Authentication
* Anything a server can do

#### Abbreviations

* Amazon Machine Images - AMIs
* Elastic Block Store - EBS
* Input-Output per second - IOPS
* Elastic File Systems - EFS

### **Amazon Machine Image**

An Amazon Machine Image (AMI) provides the information required to launch an instance. You must specify an AMI when you launch an instance.

#### An AMI includes:

1. One or more EBS snapshots, or, for instance-store-backed AMIs.(Used for backup)
2. A template for the root volume of the instance (for example, an operating system, an application server, and applications).
3. Launch permissions that control which AWS accounts can use the AMI to launch instances.
4. A block device mapping that specifies the volumes to attach to the instance when it's launched.

#### Managing AMI’s

1. Pre-Built
2. AWS Marketplace
3. Create your own

#### Significance of AMI’s

* Repeatability
* Reusability
* Recoverability
* Marketplace Solutions
* Backups

### **EBS Volumes**

Amazon Elastic Block Store (Amazon EBS) provides block-level storage volumes for use with EC2 instances. EBS volumes behave like raw, unformatted block devices.

* Applications need block-level storage, Instance store is ephemeral, Need data to persist through shutdowns, and need to be able to backup data volumes are the problems. So we use EBS Volumes to overcome these problems.
* EBS functions as a hard disk.

To create a snapshot for Amazon EBS volumes that serve as root devices you should stop the instance before taking this snapshot.

However, you can take a snap while the instance is running.

You can create AMI from EBS-backed instances and snapshots.

You can change EBS volume sizes on the fly, including changing the size and storage types. Volumes will always be in the same availability zones as the ec2 instance.

### **INSTANCE STORE VOLUMES**

Unlike EBS volumes, instance store volumes are ephemeral. This means that when the instances they are attached to shut down, their data is permanently lost.

Instance store volumes work well for deployment models where instances are launched to fill short-term-goals (as part of audio scaling groups, for instance), import data from external sources, and are effectively disposable

**At a single instance**, **a single EBS volume is associated with a single EC2 instance only, but a single EC2 instance can have any number of EBS Volumes**

#### ****EBS Volume**** Types ****Solid**** State-Backed

1. **General Purpose SSD -**General Purpose SSD volume that Balances price and performance for a wide variety of workloads
   1. **Use cases**
      1. Recommended for most workloads
2. **Provisioned IOPS SSD -** High-Performance SSD volume for mission-critical low latency or high throughput workloads
   1. **Use cases**
      1. Critical business applications that require sustained IOPS performance
      2. Large database workloads
      3. Throughput Optimized HDD

#### ****EBS Volume types Hard Disk Backed****

1. **Low-Cost HDD -** Low-cost HDD volume designed for frequently accessed throughput intensive workloads
   1. **Use cases**
      1. Streaming workloads big data
      2. data warehouses
      3. log processing
      4. cannot be a boot volume
2. **Cold HDD -** Lowest cost HD volume designed for less frequently accessed workloads
   1. **Use cases**
      1. throughput oriented Storage for large volumes of data that is infrequently accessed
      2. scenarios where the lowest storage cost is important
      3. cannot be a boot volume

**Solid State Drive(SSD) is faster than Hard Disk(HD) because HD has read-write assembly and SSD can be used as a root volume Whereas HDD’s can’t be used**

### **Shared file System**

If I have multiple instances that need to use the same storage. Amazon EBS only attached to one instance whereas Amazon S3 is an option but it is not ideal and Amazon EFS and Amazon FSx are perfect for this task

### **Amazon Elastic File Systems(EFS)**

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources.

### **Instance Types Naming**

#### m5.large

* M is the family name
* 5 is the generation number
* Large is the size of the instance

### **EC2 instance types**

* general-purpose
  + good for burstable workloads like website and web applications
* compute-optimized
  + optimized compute-intensive workloads.
  + workloads requiring significant processing
* memory-optimized
  + memory heavy applications or when you need more Ram than CPU. For memory-intensive workloads
* accelerated computing
  + performance GPU-based instances
  + Commonly used for Machine learning or Deep learning
* storage optimized
  + Up to 16tb of his HDD local storage with high disk throughput.
  + For workloads requiring high amounts of fast SSD storage

GPU based instances intended for graphics and general-purpose GPU compute workloads.

### **Instance lifecycle**

* The state of running ec2 instance can be managed in a number of ways
* Terminating the instance will shut it down and cause its resources to be reallocated to the general AWS pool
* Stop and restart
  + Impact on instance store volumes of data and IP address assigned

### **Security Groups**

* A security group is a virtual Firewall.
* It controls the traffic for the instances.
* One instance can have multiple security groups
* You can have any number of ec2 instances with in security group and vice versa
* All inbound traffic is blocked by default so you have to create rules in the security group

#### ****LifeCycle****

1. Choose AMI
2. Choose Instance Type
3. Configuring Instance  Details
4. Add Storage
5. Add Tags
6. Configure Security Group(Virtual Firewalls)
7. Review

### **Lab Sessions**

### **Creating and Managing a Windows Virtual Machine**

**Creating Windows Instance**

1. Launch Instance in the EC2 dashboard
   1. Launch instance
   2. Select Microsoft Windows server 2012
2. Selected AMI
3. Selected Instance type
4. Configure Instance Details
   1. Launch any number of Instances
   2. Default vpc
   3. Shutdown behavior
5. Add storage
6. Add tags
7. Configure Security Groups
8. Review
9. Launch

**Launching the server**

1. Select instance & click on connect in actions
2. In RDP client
   1. Download remote desktop file
   2. Click on Get Password
   3. Browse  the key pair - letsupgrade123.pen
   4. Decrypt the password
   5. Copy the password
   6. Open the RDP file that is downloaded
   7. Click on connect
   8. A pop will be appeared and place the password in Administrator
   9. And click ok

### Creating and launching Linux machine

#### Download MobaXterm application file

* Google MobaXterm download
* MobaXterm Xserver with ssh, telnet
* Go for home edition and download the portable edition

**Creating a LINUX Instance**

1. Select first AMI
2. Choose an Instance type -1 instance
3. Configure instance details
4. Add storage
5. Add tags
6. Configure security group
7. Select keypair and launch

**Launching a LINUX Instance**

1. Select server
2. Actions -> Connect
3. In EC2 Instance Connect
   1. Copy public IP address
4. Open MobaXterm
   1. Click on session
   2. Click on SSH
   3. In remote host - PASTE IP address
   4. In Specify username ec2-user
   5. In advanced SSH settings
      1. Use private key
         1. Browse - letsupgrade123.pen
      2. Click on OK
5. Click on connect

**Terminating the instances**

* In services
  + Click on EC2
    - Click on Running instances
      * Click on server
      * Actions
        + Instance state
        + Terminate instance

Terminate

If failed

Failed to terminate the instance

Modify the disable termination

Actions

Instance settings

Change Termination protection

Untick Enable and save

Successfully done

## Day 3 Agenda Total Duration is 3:03:38

* Revision of Days 1 and 2
* Creating Windows and Ubuntu Instances
* Theory of EBS volumes
* Introduction to storage volumes
* Working with volumes

### Creating a Windows Instance:

1. Launch Instance in the EC2 dashboard
   1. Launch instance
   2. Select Microsoft Windows server
2. Selected AMI
3. Selected Instance type
4. Configure Instance Details
   1. Launch any number of Instances
   2. Default vpc
   3. Shutdown behavior
5. Add storage
6. Add tags - Tagging increases the visibility
7. Configure Security Groups - Anywhere - instance is open to any traffic and from any location and any service
8. Review
9. Launch

Ensure a 2/2 check, before connecting a machine. The 2/2 check is not properly checked then the hardware and software connection is not properly made and results in some errors.

**Launching the server**

1. Select instance & click on connect in actions
2. In RDP client
   1. Download remote desktop file
   2. Click on Get Password
   3. Browse  the key pair - letsupgrade123.pen
   4. Decrypt the password
   5. Copy the password
   6. Open the RDP file that is downloaded
   7. Click on connect
   8. A pop will be appeared and place the password in Administrator
   9. And click ok

**Powershell prompt**

* Find the PowerShell prompt in the virtual machine
* In C:\Users\Administrators> Install-WindowsFeature -name Web-Server -IncludeManagementTools
* press enter
* In EC2 > Instances > Public IPv4 address > copy-paste the address in web browser

Public IP - independent or individual instances

In EC2 > Instances > Public IPv4 DNS -

**Questions**

* What happens to the installed applications when I restart my machine/stop and start a machine?
* Will I be able to access the application using the same public IP?

**Ubuntu Server 20.04LTS**

1. Select first AMI
2. Choose an Instance type -1 instance
3. Configure instance details
4. Add storage
5. Add tags
6. Configure security group
   1. All traffic
   2. Custom
      1. Anywhere
7. Select keypair and launch

We can add any group in security groups if we have more traffic in the

1. configure security group
   1. Enabling SSH
   2. Add rule
      1. HTTPS
      2. POP3

ARN - Amazon Resource Number

Amazon Resource Names (ARNs) uniquely identify **AWS** resources. We require an **ARN** when you need to specify a resource unambiguously across all of **AWS**, such as in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls.

**EC2**

* Security Groups
  + ARN
    - Edit Inbound rules
      * Add type(maybe All traffic or SSH etc)
      * Save rules (Adding rules)

**Launching a UBUNTU Instance**

1. Select server
2. Actions -> Connect
3. In EC2 Instance Connect
   1. Copy public IP address
4. Open MobaXterm
   1. Click on session
   2. Click on SSH
   3. In remote host - PASTE IP address
   4. In Specify username ec2-user
   5. In advanced SSH settings
      1. Use private key
         1. Browse - letsupgrade123.pen
      2. Click on OK
5. Click on connect
6. In Ubuntu Virtual machine
   1. sudo apt-get -y update
      1. Click enter
   2. sudo apt-get -y install nginx
      1. Click enter
7. Copy-paste public IP in a web browser

**Terminating the instances**

* In services
  + Click on EC2
    - Click on Running instances
      * Click on server
      * Actions
        + Instance state
        + Terminate instance

Terminate

If failed

Failed to terminate the instance

Modify the disable termination

Actions

Instance settings

Change Termination protection

Untick Enable and save

Successfully done

**IAM ROLES**

* You can also control access to AWS resources—including EC2 instances—through the Use of IAM roles.
* You define an IAM role by giving it permissions to perform actions on Specified services or resources within your AWS account.
* When a particular role is assigned To a user or resource, they’ll gain access to whichever resources were included in the role policies.

**NAT DEVICES**

For example, Lets I have a webserver in a public subnet and a Database in a private subnet in a VPC. enables private IP networks that use unregistered IP addresses to connect to the Internet. **NAT** operates on a router, usually connecting two networks together, and translates the private (not globally unique) addresses in the internal network into legal addresses before packets are forwarded to another network.

You can use a network address translation (NAT) instance in a public subnet in your VPC to enable instances in the private subnet to initiate outbound IPv4 traffic to the internet or other AWS services, but prevent the instances from receiving inbound traffic initiated by someone on the internet.

**KEY PAIRS**

* To ensure properly secured sessions, you’ll need to generate a key pair, save the public key to your EC2 server, and save its private half to your local machine.
* If you’re working with a windows AMI, you’ll use the private key file to retrieve the password you’ll need to authenticate into your instance.
* For a Linux AMI, the private key will allow you to open an ssh session.
* Each key pair that was generated for you will remain installed within its original region and available for use with newly launched instances until you delete it.
* You should delete the AWS copy in the event your public key is lost or exposed. Just be careful before you mess with your keys: your access to an instance might depend on it.

#### EC2 PRICING OPTIONS



**1. On-demand Instances**

On-Demand Instances let you pay for computing capacity by the hour or second (minimum of 60 seconds) with no long-term commitments. This frees you from the costs and complexities of planning, purchasing, and maintaining hardware and transforms what are commonly large fixed costs into much smaller variable costs.

**2. Reserved Instances**

Amazon EC2 Reserved Instances (RI) provide a significant discount (up to 72%) compared to On-Demand pricing and provide a capacity reservation when used in a specific Availability Zone.

**Types**

With RIs, you can choose the type that best fits your applications needs.

* Standard RIs: These provide the most significant discount (up to 72% off On-Demand) and are best suited for steady-state usage.
* Convertible RIs: These provide a discount (up to 54% off On-Demand) and the capability to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or greater value. Like Standard RIs, Convertible RIs are best suited for steady-state usage.
* Scheduled RIs: These are available to launch within the time windows you reserve. This option allows you to match your capacity reservation to a predictable recurring schedule that only requires a fraction of a day, a week, or a month.

**3. Saving plans**

Savings Plans is a flexible pricing model that provides savings of up to 72% on your AWS compute usage. This pricing model offers lower prices on Amazon EC2 instances usage, regardless of instance family, size, OS, tenancy, or AWS Region, and also applies to AWS Fargate and AWS Lambda usage.

Savings Plans offer significant savings over On-Demand, just like EC2 Reserved Instances, in exchange for a commitment to use a specific amount of computing power (measured in $/hour) for a one or three-year period.

**4. Spot Instances**

Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS cloud. Spot Instances are available at up to a 90% discount compared to On-Demand prices. You can use Spot Instances for various stateless, fault-tolerant, or flexible applications such as big data, containerized workloads, CI/CD, web servers, high-performance computing (HPC), and test & development workloads. Because Spot Instances are tightly integrated with AWS services such as Auto Scaling, EMR, ECS, CloudFormation, Data Pipeline, and AWS Batch, you can choose how to launch and maintain your applications running on Spot Instances.

Moreover, you can easily combine Spot Instances with On-Demand, RIs, and Savings Plans Instances to further optimize workload cost with performance. Due to the operating scale of AWS, Spot Instances can offer the scale and cost savings to run hyper-scale workloads. You also have the option to hibernate, stop, or terminate your Spot Instances when EC2 reclaims the capacity back with two-minutes of notice. Only on AWS, you have easy access to unused compute capacity at such a massive scale - all at up to a 90% discount.

#### ****Dedicated options****

**1. Dedicated instances**

Dedicated instances are physically isolated from the other AWS accounts

**2. Dedicated Hosts**

Amazon EC2 Dedicated Hosts allow you to use your eligible software licenses from vendors such as Microsoft and Oracle on Amazon EC2 so that you get the flexibility and cost-effectiveness of using your own licenses, but with the resiliency, simplicity, and elasticity of AWS. An Amazon EC2 Dedicated Host is a physical server fully dedicated for your use, so you can help address corporate compliance requirements.

Amazon EC2 Dedicated Host is also integrated with AWS License Manager, a service that helps you manage your software licenses, including Microsoft Windows Server and Microsoft SQL Server licenses. In License Manager, you can specify your licensing terms for governing license usage, as well as your Dedicated Host management preferences for host allocation and host capacity utilization. Once set up, AWS takes care of these administrative tasks on your behalf, so that you can seamlessly launch virtual machines (instances) on Dedicated Hosts just like you would launch an EC2 instance with AWS provided licenses.

#### Placement Groups

**1. Cluster placement groups**

* The cluster placement group is a logical grouping of instances within a single Availability Zone.
* This grouping provides the lowest latency and highest packet per second network performance possible.

**2. Spread Placement Groups**

* A spread placement group is a grouping of instances that are purposely positioned on the distinct underlying hardware.
* This grouping reduces the risk of simultaneous failures that could occur if instances were sharing underlying hardware.
* This type of group can span multiple Availability Zones, up to a maximum of seven instances per Availability Zone per group.

**3. Partition Placement group**

* A spread placement group is a grouping of instances that are purposely positioned on the distinct underlying hardware.
* This grouping reduces the risk of simultaneous failures that could occur if instances were sharing underlying hardware.
* This type of group can span multiple Availability Zones, up to a maximum of seven instances per Availability Zone per group.

### AWS Elastic BeanStalk

* Elastic beanstalk lets you upload your application code and define a few parameters, and AWS will configure, launch, and maintain all the infrastructure necessary to keep it running.
* That might include EC2 load-balanced and autoscaled instances, RDS database instances, and all the network plumbing you would otherwise have had to build yourself.
* Compatible languages and platforms include .NET, java, node.Js, python, and docker. Elastic beanstalk adds no charges beyond the cost of the running infrastructure itself.

### Lab3 - Storage volumes

Creating a Windows Instance:

1. Launch Instance in the EC2 dashboard
   1. Launch instance
   2. Select Microsoft Windows server 2012
2. Selected AMI
3. Selected Instance type
4. Configure Instance Details
   1. Launch any number of Instances
   2. Default vpc
   3. Shutdown behavior
5. Add storage
6. Add tags
7. configured Security Groups
8. Review
9. Launch

#### Creating storage volume

1. Elastic Block Store
2. Volumes
3. Create Volume
4. Size - 2
5. Availability - us east 2 a
6. Actions
7. Attach volume
8. Click Ok

**Launching the server**

1. Select instance & click on connect in actions
2. In RDP client
   1. Download remote desktop file
   2. Click on Get Password
   3. Browse  the key pair - letsupgrade123.pen
   4. Decrypt the password
   5. Copy the password
   6. Open the RDP file that is downloaded
   7. Click on connect
   8. A pop will be appeared and place the password in Administrator
   9. And click ok

In the virtual machine

1. Go to server manager
   1. File and storage devices
   2. Volumes
   3. Disks
   4. Select volume
      1. Right-click new volume
      2. Bring online
      3. Next -> next ….. Finish
2. File Explorer
   1. C and D drives are available

Terminating Instances

* In services
  + Click on EC2
    - Click on Running instances
      * Click on server
      * Actions
        + Instance state
        + Terminate instance

Terminate

Successfully done

Now these volumes will be available

* Actions
* Modify volume
* Can’t change the availability zone but can change the size and type

**Creating a snapshot**

* Volumes
* Create snapshot
* Description
  + Give a Name
  + Click on Create snapshot

**Deleting a volume**

* Click on volume
* Actions
* Delete volume
* Click on Yes, Delete

Delete the snapshots, volumes, Instances

#### Working with volumes

* Create a windows machine
* Create a volume in the same region as  the window machine
* Attach the volume in the windows machine
* From Server Manager bring the volume inline
* once the ebs is online create a new volume
* check if the volume is mounted successfully
* Try modifying the volume config
* delete the volume

## Day 4 Agenda - Total duration is 3:07:09

* Working with Elastic IP
* Working with Amazon web services S3
* Static web hosting
* Versioning
* Storage Classes and Data Lifecycle

## Working with Elastic IP - Lab

**What is an Elastic IP?**

               An **Elastic IP** address is a reserved public **IP** address that you can assign to any EC2 instance in a particular region until you choose to release it. To allocate an **Elastic IP** address to your account in a particular region, see Allocating an **Elastic IP** Address. It is a static paid public IP.

**Creating a Linux instance**

1. Select first AMI
2. Choose an Instance type -1 instance
3. Configure instance details
4. Add storage
5. Add tags
6. Configure security group
7. Select keypair
8. launch

**Launching a LINUX Instance**

1. Rename the project “apache”
2. Ensure a 2/2 status check
3. Select server
4. Actions -> Connect
5. In EC2 Instance Connect
   1. Copy public IP address
6. Click on connect

**Working with elastic IPs**

1. Install an Apache server
   1. Switch to the root user
   2. sudo -s
2. Now run the updates
   1. yum -y update
3. Once completed, install and run the apache server
4. Install an apache webserver
   1. yum install httpd
   2. When prompted, press “Y” to confirm
5. Start the webserver
   1. systemctl start httpd
6. Enable httpd
   1. systemct1 enable httpd
7. Check web server status
   1. systemct1 status httpd
8. You can see the active status is running
9. Copy-paste the public IP address in a web browser
10. You will receive an “Apache TEST PAGE”
11. EC2 Dashboard
12. Network and Security
    1. Elastic IPs
       1. Allocate Elastic IP address
          1. Click on allocate
       2. Rename the elastic IP as ” EIPdemo”
       3. Select IP
          1. Actions
             1. Associate Elastic IP address

Instance(here EC2)

Choose an instance(apache)

Select instance id of running instance

Tick allows this elastic IP address to be reassociated

Click on Associate

* EC2 Dashboard
* Instances
* Apache instance
* Open the web page using the elastic IP

Elastic IPs are chargeable if it is not associated

**Terminating the instances**

* In services
  + Click on EC2
    - Click on Running instances
      * Click on server
      * Actions
        + Instance state
        + Terminate instance

Terminate

If failed

Failed to terminate the instance

Modify the disable termination

Actions

Instance settings

Change Termination protection

Untick Enable and save

Successfully done

**Terminating the Elastic IP**

1. EC2 Dashboard
2. Network and Security
   1. Elastic IPs
      1. Release Elastic IP address
      2. Click on Release
3. IP will be moved to a shared pool

**Disassociate the Elastic IP**

1. EC2 Dashboard
2. Network and Security
   1. Elastic IPs
      1. Disassociate Elastic IP address
      2. Click on Dissociate

## AWS Storage

The **AWS Storage** Gateway is a hybrid **storage** service that enables your on-premises applications to seamlessly use **AWS** cloud **storage**. You can use the service for backup and archiving, disaster recovery, cloud data processing, **storage** tiering, and migration.

## AWS S3 - SIMPLE STORAGE SERVICES

* S3 is object-based storage i.e allows you to upload files.
* .exe files cant be stored
* Files can be from 0 bytes to 5tb
* There is unlimited storage.
* Files are stored in buckets ( folder)
* S3 is universal namespaces, that is names must be unique globally.
  + Global
    - Unique name
* [Https://s3-eu-west-1.Amazonaws.Com/bucketname](https://s3-eu-west-1.amazonaws.com/bucketname)
* S3- 99.99% availability, 99.9999…11’s 9 durable.

### AWS S3 - SIMPLE STORAGE SERVICE TIERS

* S3 – IA ( infrequently accessed ): for the data that is accessed less frequently, but requires rapid access when needed. Lower fee than S3.
* The glacier is an extremely low-cost storage service for data archival
* Glacier stores data for a little as $0.01 per GB per month and is optimized for data that is infrequently accessed and for which retrieval times of 3-4 hrs.

### **S3 PRICING**

S3 charged for

* Storage
* Requests ( no of times requests made for bucket access )
* Storage management pricing
* Data transfer pricing
* Transfer acceleration

### Object Storage in S3

**Object Storage in S3- lab**

* Open Amazon S3
  + Click on create a bucket
  + Give a name for the bucket
  + Choose any region
  + Create
  + Choose chris123456 bucket
    - Click on upload
    - Add files
    - Upload a file
    - Click on upload
  + Open file
    - Copy object Url
  + Paste it in the browser
    - Access Denied
  + Inside bucket
    - Click on the Permissions tab
    - Click on edit
      * Uncheck block all public access
        + Click on save
      * Type confirm
        + Click on confirm
    - Click on the overview tab
      * Click on make public
    - Now access the image using the URL(object URL)
  + Select object
    - Details about the object are displayed
    - We have download, copy path options
    - Change the storage class
    - Change encryption
    - Add tags
    - Rename
    - Delete

### **S3 Access points**

Amazon **S3 Access Points**, a feature of **S3**, simplifies managing data **access** at scale for applications using shared data sets on **S3**. **Access points** are unique hostnames that customers create to enforce distinct permissions and network controls for any request made through the **access point**

They support

* Single user
* Applications
* Groups of users or applications

### **Static web hosting**

**Static web hosting - Lab**

* Go into chris123456 bucket
  + Click on properties
    - Click on static web hosting
    - Use this bucket to host a website
    - Create two text file
    - Index.txt and error.txt
      * Open text file
      * Write into the file and Save
    - Type the names of text files
    - Click on save
  + Go to the overview tab
    - Upload those text files and make public
  + Go to the properties tab
    - Static web hosting
    - Copy the endpoint and paste in a web browser

### Versioning in AWS S3

**Versioning** is a means of keeping multiple variants of an object in the same bucket. You can use **versioning** to preserve, retrieve, and restore every **version** of every object stored in your Amazon **S3** bucket. **Versioning**-enabled buckets enable you to recover objects from accidental deletion or overwrite.

**Versioning - Lab**

* Go into chris123456 bucket
  + Click on properties
    - Click on Versioning
      * Create a text file
        + versionDemo.txt
      * Open text file
        + Write into the file and Save
      * Type the names of text files
        + Click on save
  + Go to the overview tab
    - Upload those text files and make public
* Create a new bucket
  + In properties
    - Enable versioning
      * Upload versionDemo file
        + Edit the content of the file
        + Upload the text file
    - Click on show versions
    - There are two versions
    - Hide the versions
    - Select the version
      * Delete the file
    - Show versions
    - Download the file
    - Upload the file

### **S3 PRICING**

S3 charged for

1. Storage
2. Requests ( no of times requests made for bucket access )
3. Storage management pricing
4. Data transfer pricing
5. Transfer acceleration

### **Storage Classes and Data Lifecycle**

            In the starting the The data is in the Amazon s3 standard. After 90 days the data is moved in to Amazon s3 infrequent Access such that if we needed that data it will take neary 1-2 hours to retrieve back.After 1 year the data is moved in to amazon glacier

**Lifecycle - Lab**

* Go into chris123456 bucket
  + Click on management
    - Add Lifecycle rule
      * Enter rule name
      * Choose rule scope
        + Limit the scope to specific prefixes or tags
        + Apply to all objects in the bucket
      * Choose next
    - Select transition
      * Transition to Standard-IA after. - 30 days
      * Transition to Intelligent-Tiering after.
      * Transition to One Zone-IA after. - 60 days
      * Transition to Glacier after.- 90 days
      * Transition to Glacier Deep Archive after.
* Don't save (If it's saved it will be charged)